

⚡️BÁO CÁO TIẾN ĐỘ THỰC TẬP⚡️

🌱Phạm Hoàng Tuấn - Trần Lê Tuyết Hương🌱

## [TUẦN 1: 23/05/2022 - 29/05/2022](https://github.com/hoangtuandev/report-vnptit)

### ✏️ *Câu 1: Kiến trúc Microservice ?*

* Một ứng dụng được chia thành nhiều “thành phần” độc lập về mặt dữ liệu, đủ nhỏ về kích cỡ và đủ độ ảnh hưởng trong hệ thống.
* Mỗi microservice có khả năng độc lập, đảm báo lắp vào hoặc tháo ra khỏi hệ thống không làm ảnh hưởng đến các thành phần khác
* Có 4 đặc trưng: micro-service, tính độc lập, tính chuyên biệt, phòng chống lỗi.

### ✏️ *Câu 2: So sánh kiến trúc Ứng dụng nguyên khối và Microservice ?*

| **Mô hình Microservice** | **Mô hình nguyên khối** |
| --- | --- |
| - Các microservice hoạt động tách biệt nhau trong hệ thống, do vậy việc build một microservice cũng độc lập với việc build các microservice khác. | - Toàn bộ ứng dụng là một khối lớn, trong khối lớn ấy có chia thành các mô đun nhỏ, mỗi mô đun thực hiện một nhiệm vụ riêng và các mô đun thường gọi nhau qua function call. |
| - Mỗi microservice là một dịch vụ chuyên biệt, có thể hoạt động độc lập. | - Việc phát triển và triển khai ứng dụng với kiến trúc này khá đơn giản khi mà các IDE hỗ trợ rất tốt việc kiểm tra và chạy ứng dụng với chỉ một cú click chuột hay một phím tắt. |
| -Kiến trúc microservice sinh ra là để dành cho các hệ thống từ lớn đến vô cùng lớn. | Kiến trúc này cũng đặc biệt phù hợp với các công ty outsource. |

### ✏️ *Câu 3: So sánh SQL Database và NoSQL Database*

|  | **CSDL quan hệ** | **CSDL không quan hệ** |
| --- | --- | --- |
| **Khối lượng công việc tối ưu** | Cơ sở dữ liệu quan hệ được thiết kế dành cho các ứng dụng xử lý giao dịch trực tuyến (OLTP). | Các cơ sở dữ liệu NoSQL được thiết kế cho các mẫu truy cập dữ liệu, bao gồm các ứng dụng có độ trễ thấp. |
| **Mô hình dữ liệu** | Mô hình quan hệ chuẩn hóa dữ liệu vào bảng được hình thành từ hàng và cột. | Các cơ sở dữ liệu NoSQL cung cấp nhiều mô hình dữ liệu khác nhau như khóa-giá trị, tài liệu và biểu đồ, được tối ưu hóa để đạt hiệu năng và quy mô tối ưu. |
| **Thuộc tính ACID** | Cơ sở dữ liệu quan hệ có các thuộc tính mang tính nguyên tố, nhất quán, tách biệt và bền vững (ACID) | Cơ sở dữ liệu NoSQL thường phải đánh đổi bằng cách nới lỏng một số thuộc tính ACID này của cơ sở dữ liệu quan hệ để có mô hình dữ liệu linh hoạt hơn có khả năng thay đổi quy mô theo chiều ngang. |
| **Hiệu năng** | Hiệu năng thường phụ thuộc vào hệ thống con của ổ đĩa. | Hiệu năng thường được xem là chức năng của kích cỡ cụm phần cứng ngầm, độ trễ mạng và ứng dụng đưa ra lệnh gọi. |
| **Quy mô** | Cơ sở dữ liệu quan hệ thường tăng quy mô bằng cách tăng năng lực điện toán của phần cứng hoặc tăng quy mô bằng cách thêm bản sao của khối lượng công việc chỉ đọc. | Cơ sở dữ liệu NoSQL thường có tính phân mảnh cao do các mẫu truy cập khóa-giá trị có khả năng tăng quy mô bằng cách sử dụng kiến trúc được phân phối để tăng thông lượng, đem đến hiệu năng ổn định với quy mô gần như không giới hạn. |
| **API** | Yêu cầu lưu trữ và truy xuất dữ liệu được truyền đạt bằng cách sử dụng các truy vấn nhất quán với ngôn ngữ truy vấn có cấu trúc (SQL). | API trên cơ sở đối tượng cho phép các nhà phát triển ứng dụng dễ dàng lưu trữ và truy xuất cấu trúc dữ liệu trong bộ nhớ. |

### ✏️ *Câu 4: Cơ sở dữ liệu phân tán*

Cơ sở dữ liệu phân tán phân tán là một tập hợp dữ liệu có liên quan (về logic) được dùng chung và phân tán về mặt vật lí trên một mạng máy tính.

### ✏️ *Câu 5: Tìm hiểu về Angular*

[**Angular**](https://angular.io/) là một Javascript Framework dùng để viết giao diện Web, xây dựng các Single Page Application(SPA)

### [Ưu điểm của Angular](https://angular.io/)

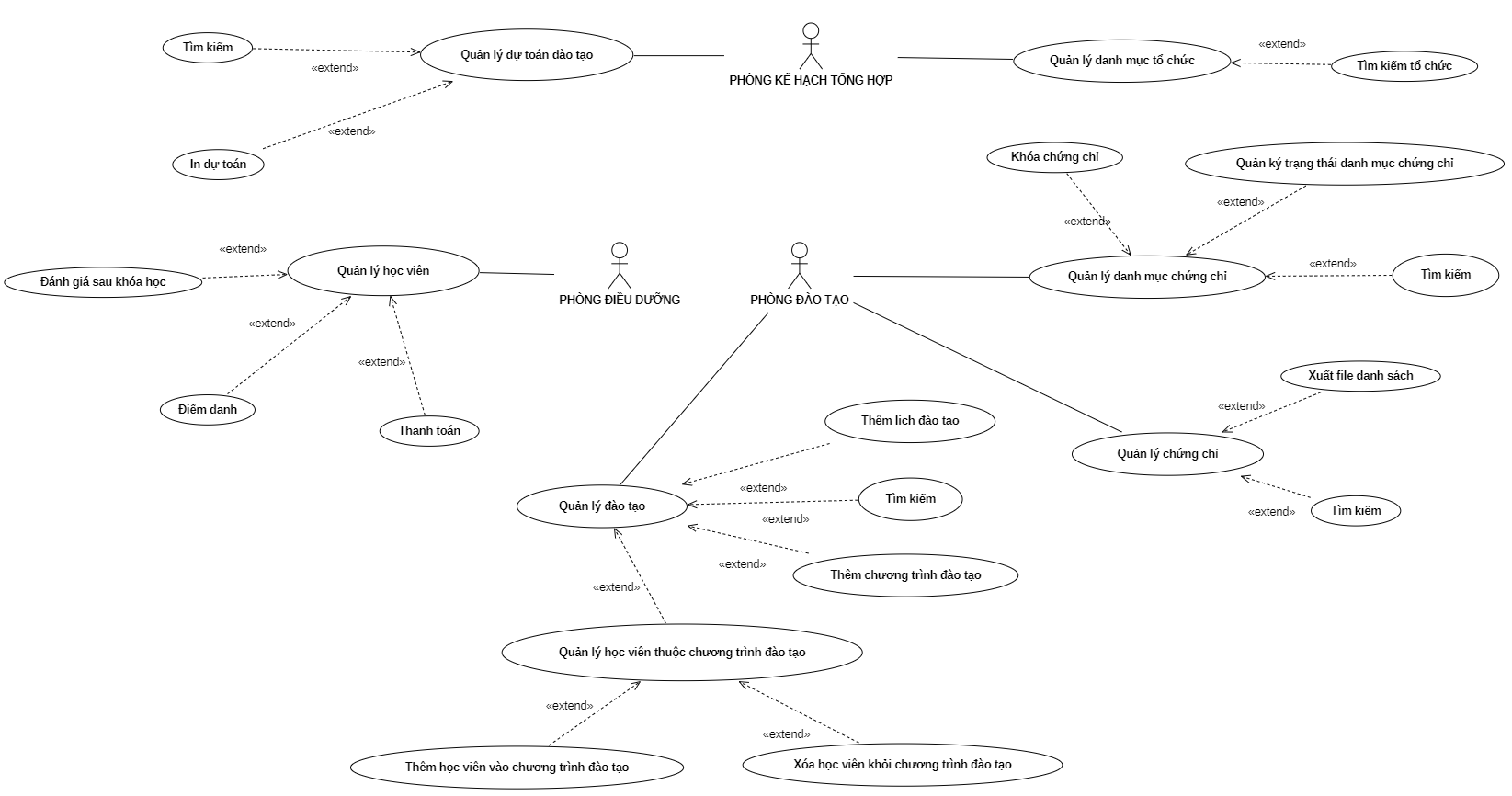
* Cú pháp mã lệnh đơn giản hơn Javascript thuần
* Phân chia layout thành các component, tăng khả năng tái sử dụng và sửa lỗi
* Tạo ra các Single Page Application (SPA).

### ✏️ *Câu 6: Tìm hiểu về Spring Boot*

* [**Spring Boot**](https://spring.io/projects/spring-boot) là một dự án phát triển bởi JAV (ngôn ngữ java) trong hệ sinh thái Spring framework. Nó giúp cho các lập trình viên chúng ta đơn giản hóa quá trình lập trình một ứng dụng với Spring, chỉ tập trung vào việc phát triển business cho ứng dụng.
* [**Spring Boot**](https://spring.io/projects/spring-boot) là một giải pháp được sử dụng rộng rãi ngày nay để phát triển ứng dụng web Java. Nó có một quy ước cố định về cách tiếp cận cấu hình. Nó hoàn toàn được điều khiển bởi cấu hình và làm cho việc sử dụng Spring Framework và nhiều thư viện của bên thứ ba khác trở nên thú vị. Các ứng dụng Spring Boot là cấp sản xuất và chỉ có thể chạy trong bất kỳ môi trường nào có cài đặt JVM. Nó tự động cấu hình Spring bất cứ khi nào có thể với các giá trị mặc định hợp lý và có POM khởi động cho nhiều mô-đun và thư viện của bên thứ ba. Nó không yêu cầu bất kỳ cấu hình XML nào và cho phép bạn tùy chỉnh các bean tự động định cấu hình bằng cách sử dụng cấu hình Java.

## [TUẦN 2: 30/05/2022 - 05/06/2022](https://github.com/hoangtuandev/report-vnptit)

### ✏️ *SƠ ĐỒ USECASE*



SoDoUseCase

🍂**1. Chức năng Quản lý danh mục tổ chức**

* Tìm kiếm tổ chức: tìm kiếm theo: mã tổ chức, tên tổ chức.

🍂**2. Chức năng Quản lý danh mục chứng chỉ**

* Tìm kiếm: tìm kiếm theo tên danh mục chứng chỉ.
* Quản lý trạng thái danh mục chứng chỉ: mỗi danh mục chứng chỉ có trạng thái hoặc “Đang sử dụng” hoặc “Ngưng sử dụng”
* Khóa chứng chỉ:
  + Thay dổi trạng thái danh mục chứng chỉ sang “Ngưng sử dụng”.
  + Đảm bảo lưu trữ chứng chỉ của nhân viên dù chứng chỉ đã ngưng sử dụng.

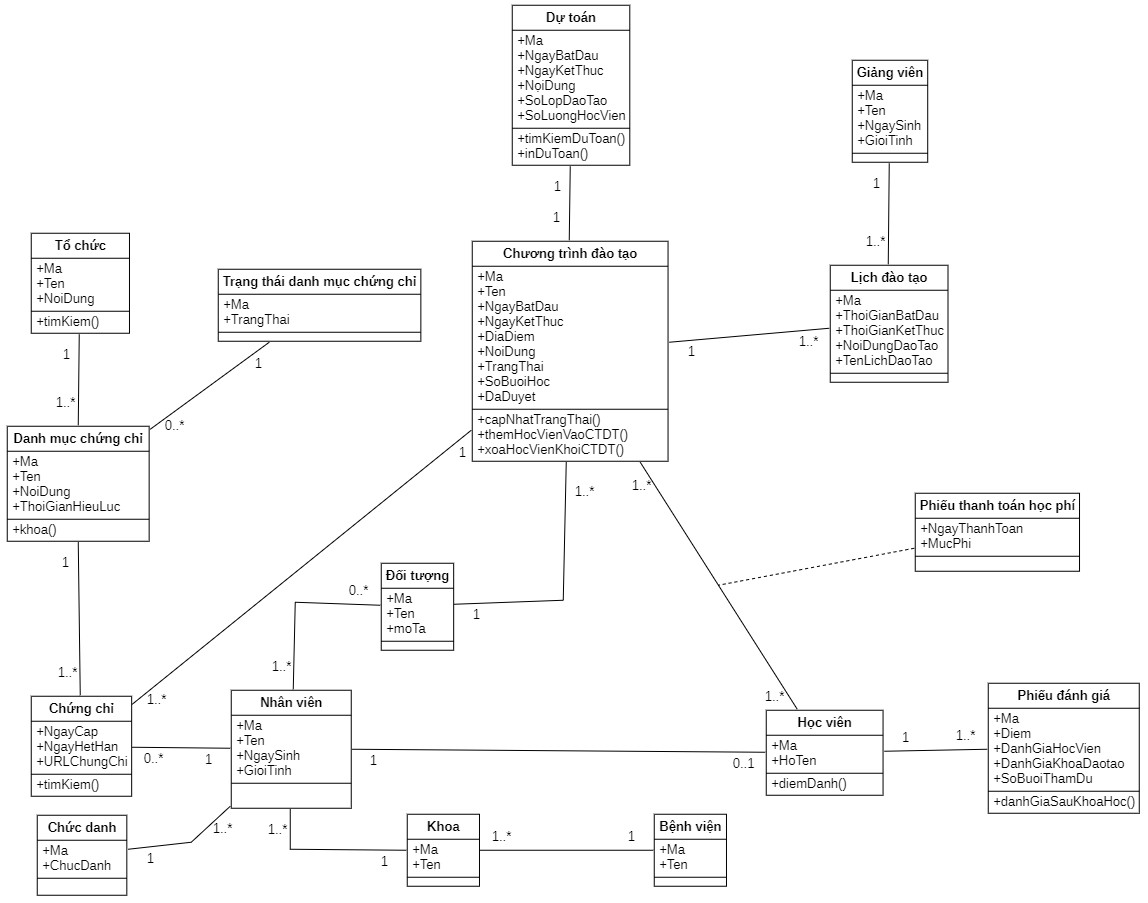
🍂**3. Chức năng Quản lý chứng chỉ** - Tìm kiếm: tìm kiếm chứng chỉ của nhân viên theo: mã nhân viên, tên nhân viên, tên đơn vị. - Xuất file danh sách: xuất file danh sách chứng chỉ của nhân viên.

🍂**4. Chức năng Quản lý đào tạo** - Thêm chương trình đào tạo. - Thêm lịch đào tạo: mỗi Chương trình đào tạo sẽ có một hoặc nhiều lịch đào tạo theo: mã đào tạo, tên đào tạo, ngày bắt đầu, ngày kết thúc, trạng thái. - Tìm kiếm: tìm kiếm chương trình đào tạo. - Quản lý học viên thuộc chương trình đào tạo: thêm học viên vào chương trình đào tạo, xóa học viên khỏi chương trình đào tạo.

🍂**5. Chức năng Quản lý học viên** - Đánh giá học viên sau khóa học. - Điểm danh: điểm danh các học viên. - Thanh toán: đánh dấu các học viên đã thanh toán học phí.

🍂**6. Chức năng quản lý dự toán** - Tìm kiếm: tìm kiếm dự toán đào tạo theo: mã dự toán, số lượng học viên. - In dự toán.

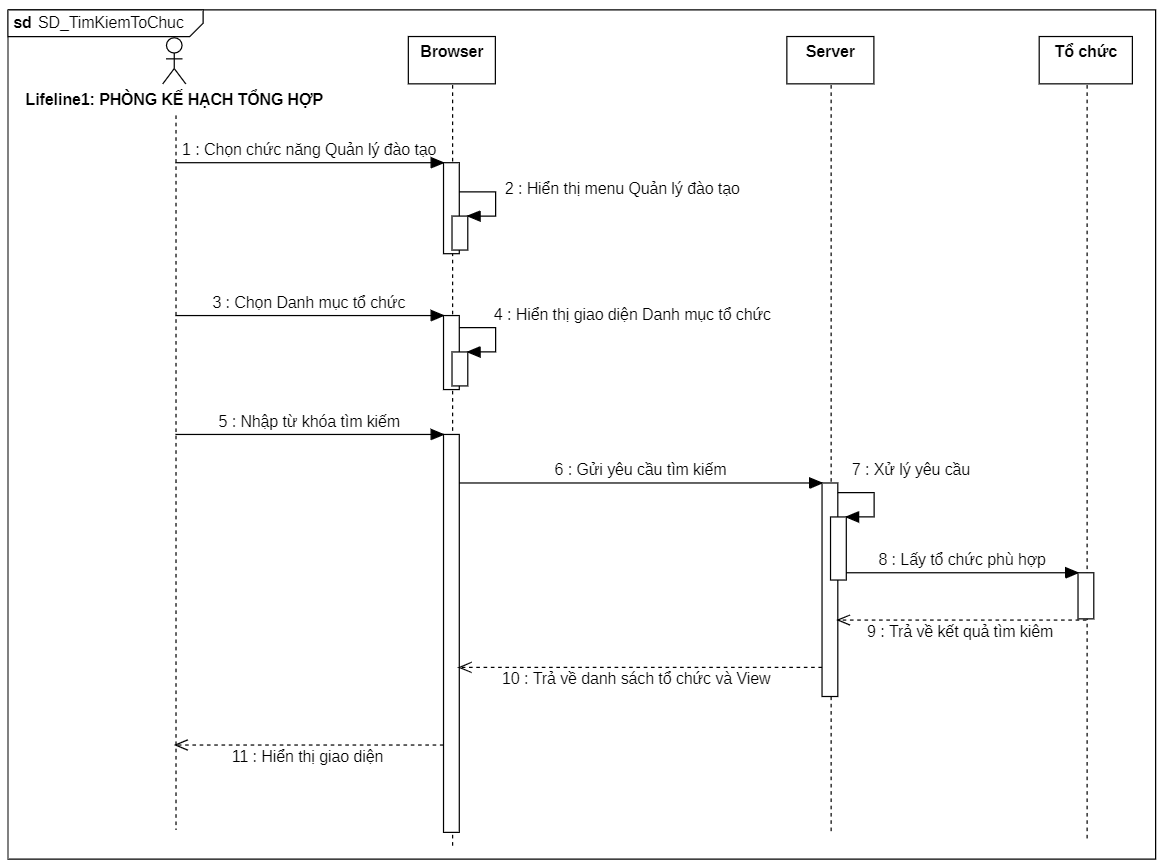
### ✏️ *SƠ ĐỒ CLASS*



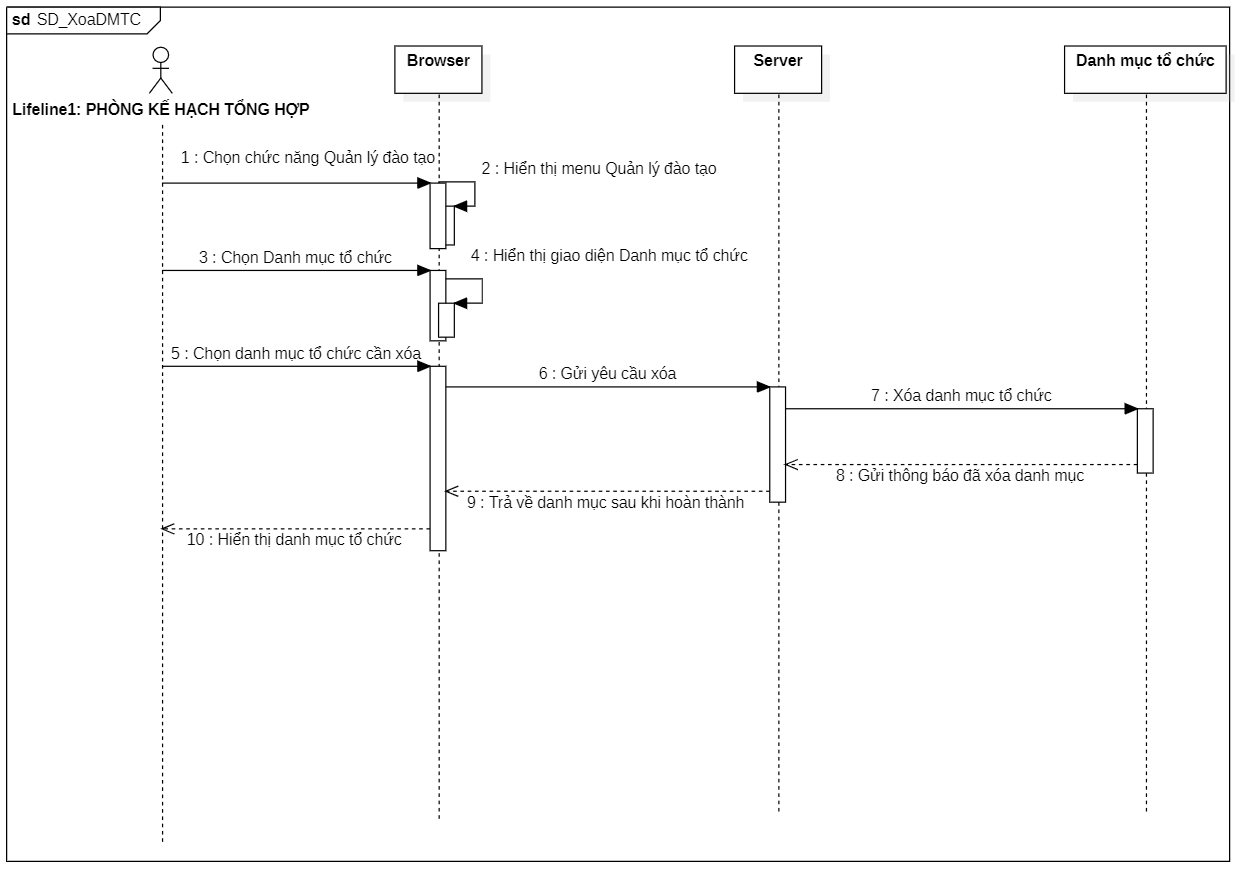
SoDoUseClass

### ✏️ *SƠ ĐỒ SEQUENCE*

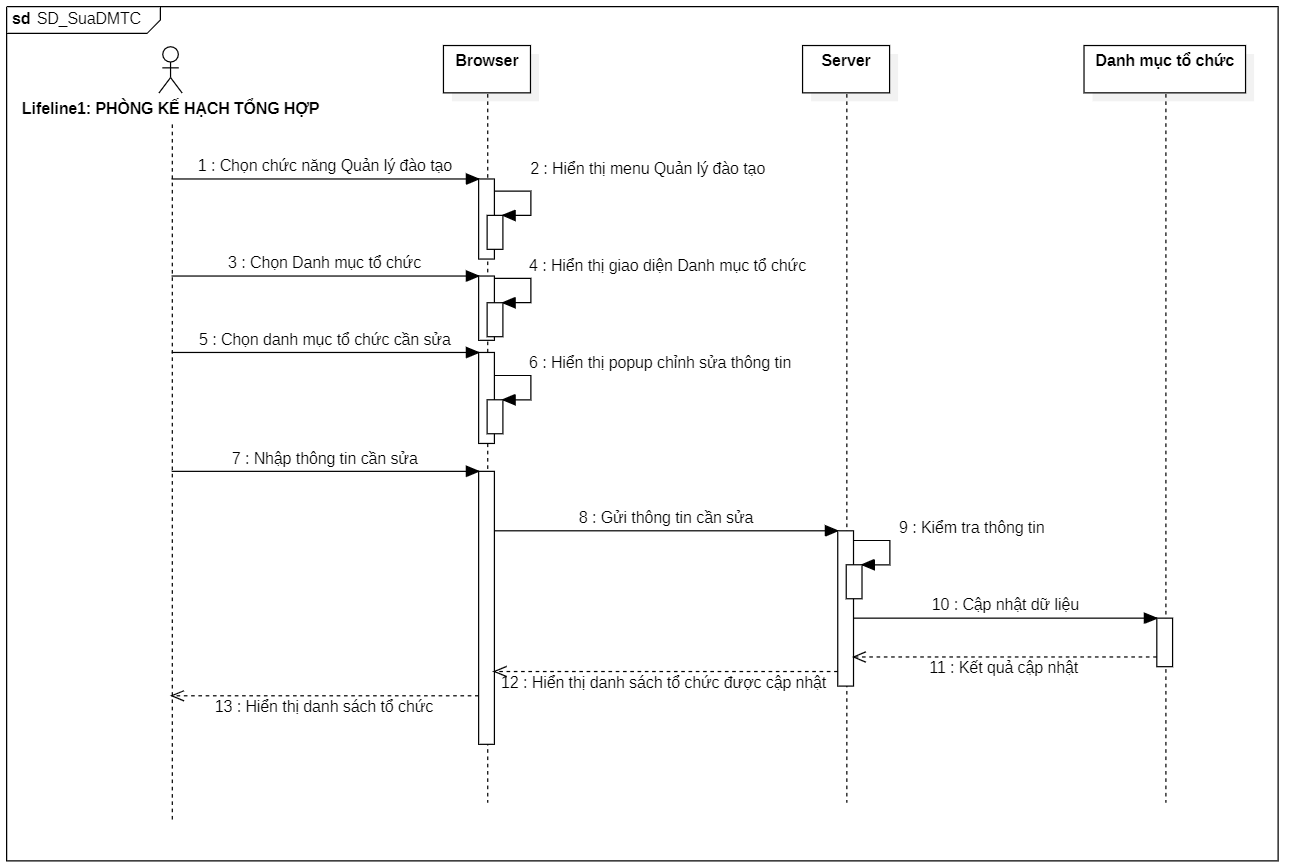
🍂**1. Tìm kiếm tổ chức**



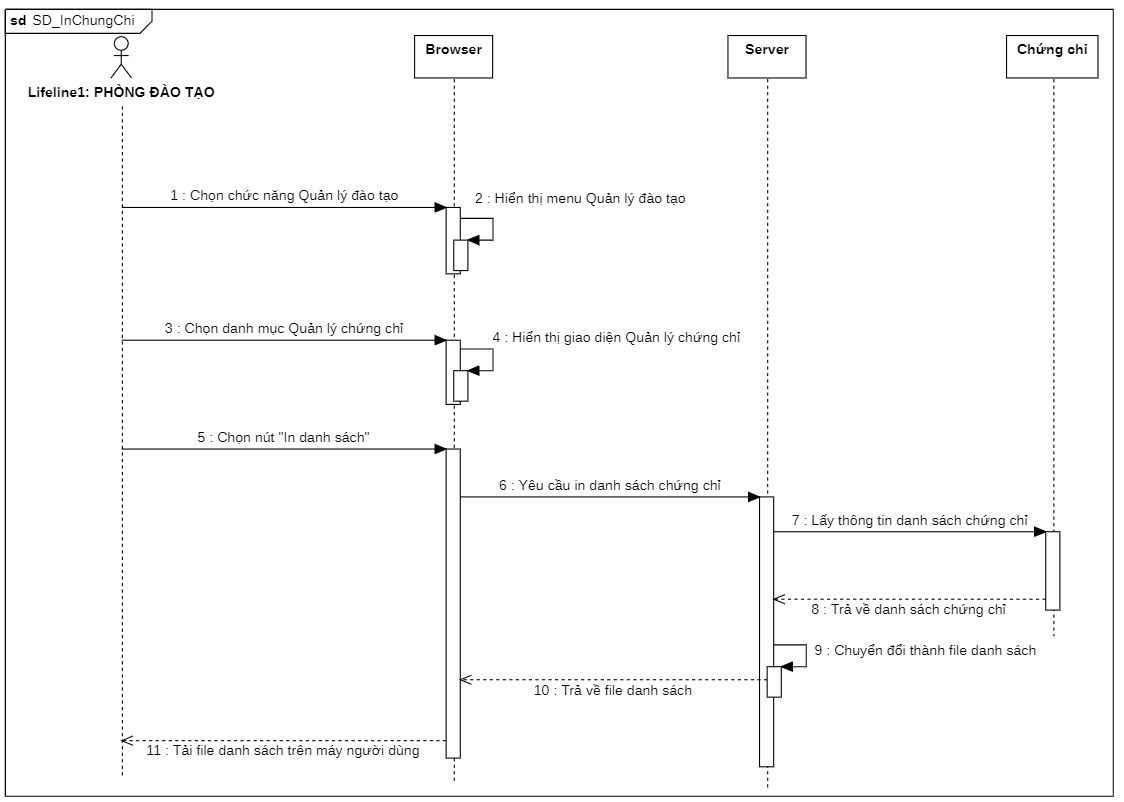
🍂**2. Xóa danh mục tổ chức**



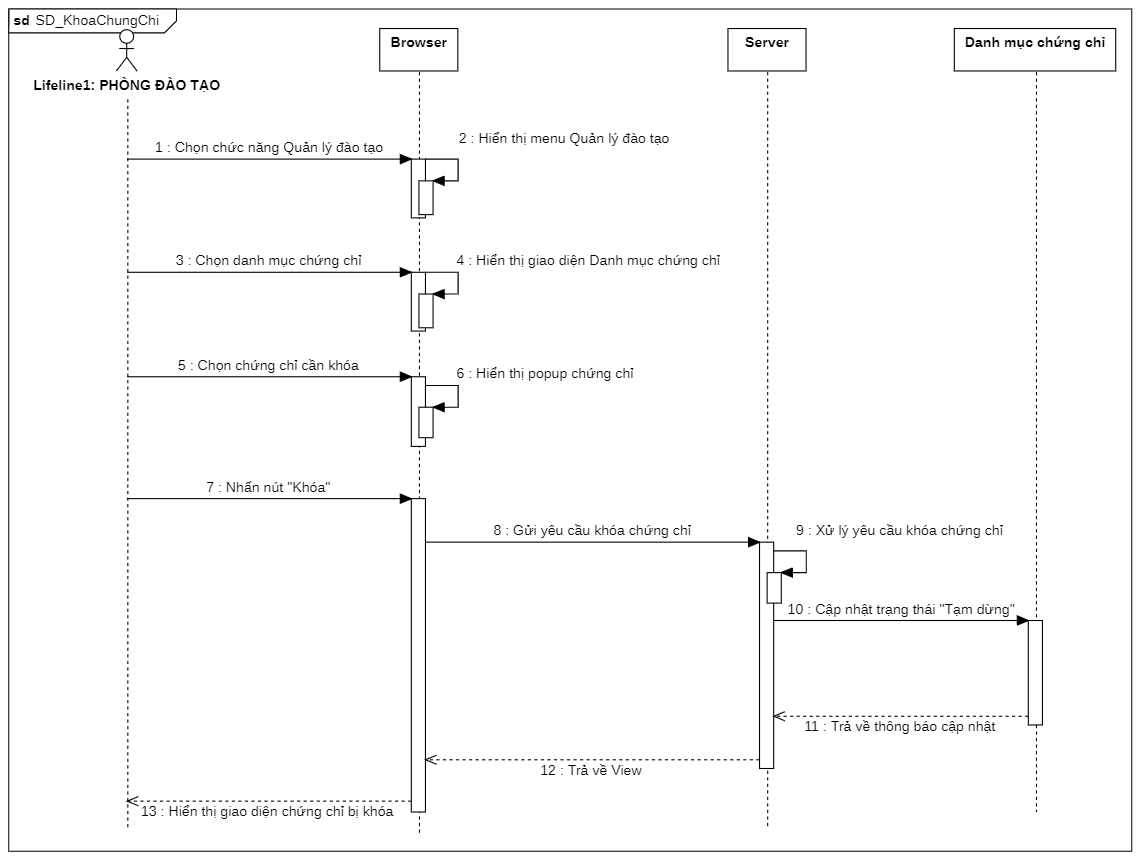
🍂**3. Sửa danh mục tổ chức**



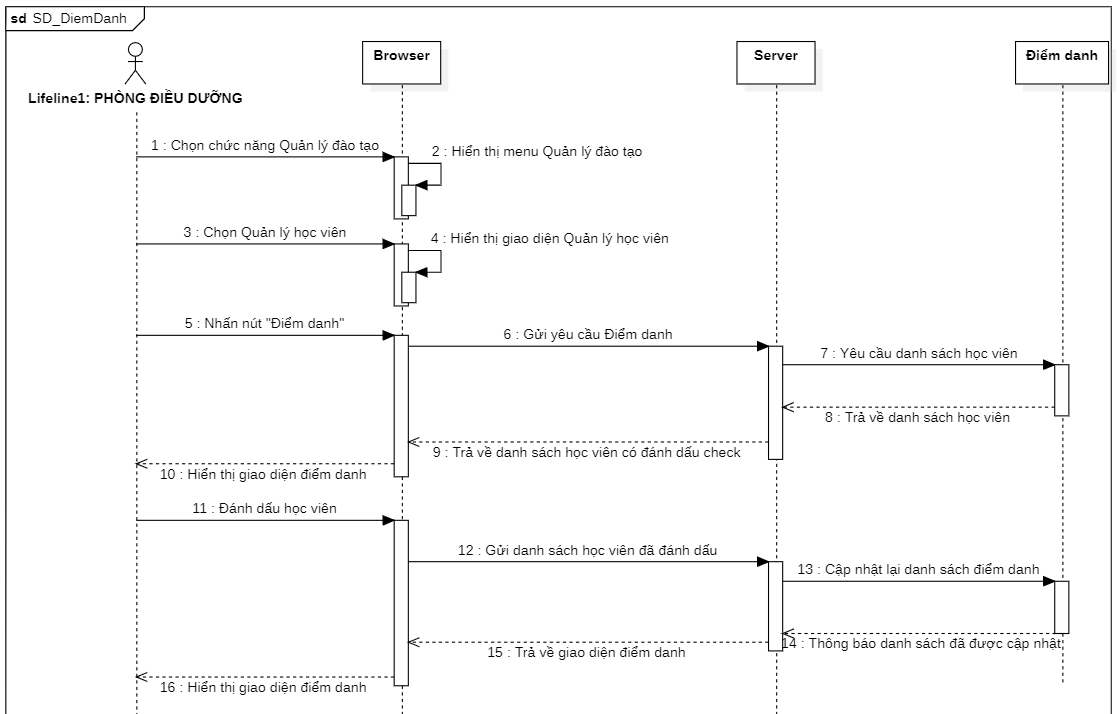
🍂**4. In chứng chỉ**



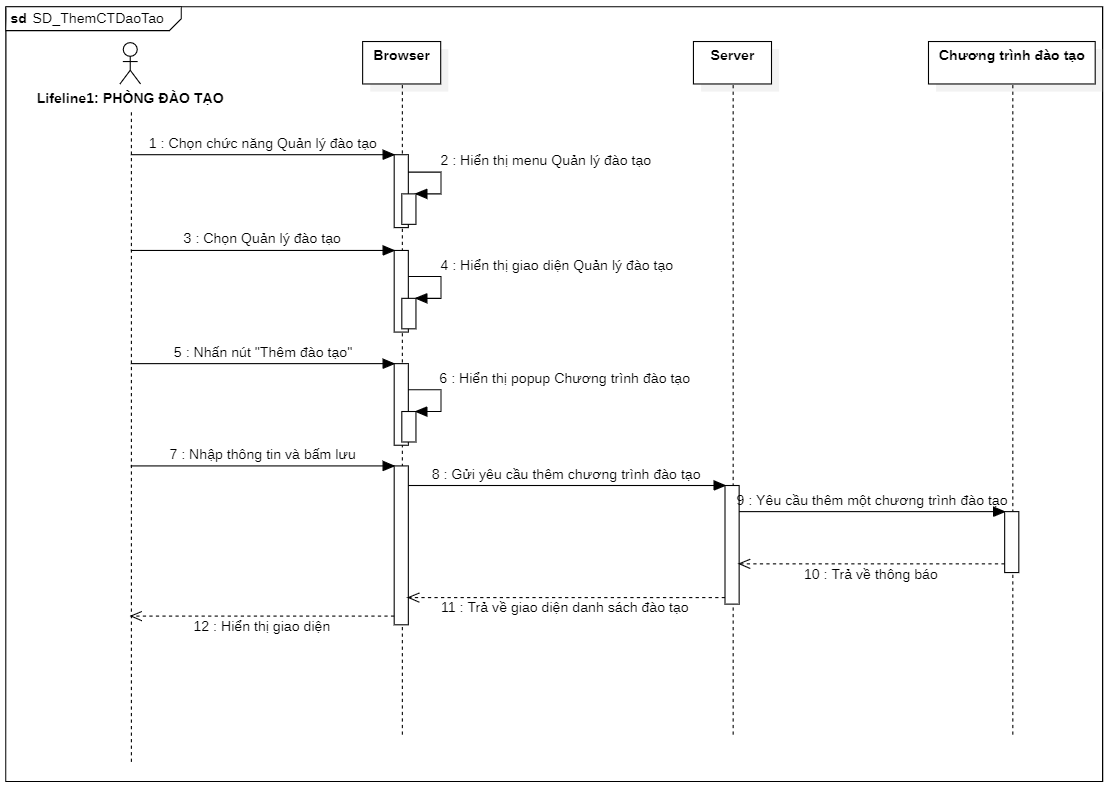
🍂**5. Khóa chứng chỉ**



🍂**6. Điểm danh**



🍂**7. Thêm chương trình đào tạo**



## [TUẦN 3: 06/06/2022 - 12/06/2022](https://github.com/hoangtuandev/report-vnptit)

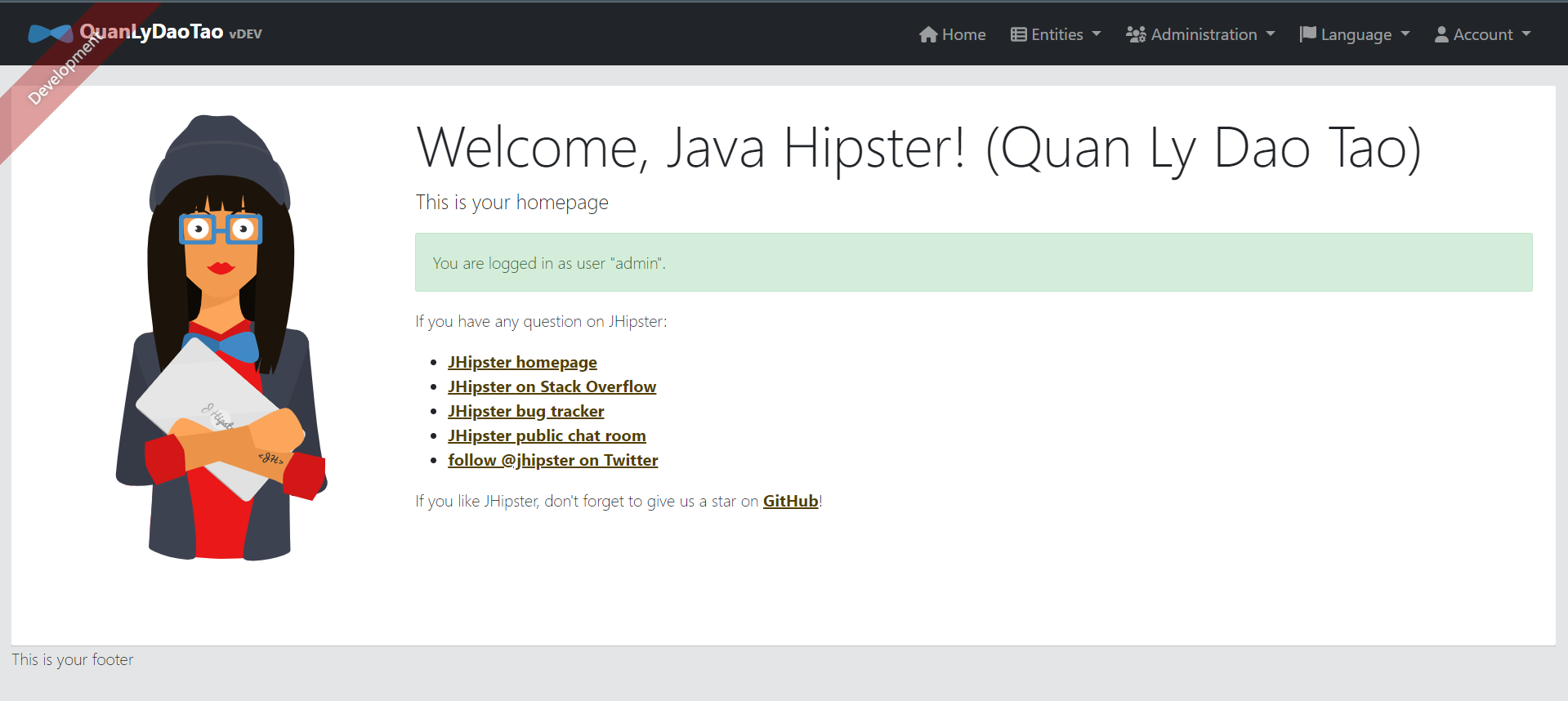
### ✏️ *Câu 1: Tìm hiểu về Jhipster*

* [**Jhipster**](https://www.jhipster.tech/) là một nền tảng để tạo, phát triển và triển khai các ứng dụng web hiện đại và kiến trúc Microservice.
* Về frontend, chúng hỗ trợ nhiều công nghệ: Angular, React và Vue. Chúng thậm chí có hỗ trợ ứng dụng mobile với React Native.
* Về backend, chúng hỗ trợ Spring Boot (với Java và Kotlin), Micronaut, Quarkus, Node.js và .NET.
* Về triển khai, chúng nắm vững các nguyên tác hoạt động của đám mây với Docker và Kubernetes. Hỗ trợ triển khai trên AWS, Azure, Cloud Foundry, Google Cloud Platform, Heroku và Open Shift.

### ✏️ *Câu 2: Angular trong dự án Jhipster*

* [**Angular**](https://angular.io/) được sử dụng để xây dựng giao diện người dụng trong dự án Jhipster
* Thư viện ng-jhipster chứa các chức năng tiện ích và các thành phần phổ biến.
* Angular CLI được sử dụng để xây dựng và thử nghiệm các ứng dụng JHipster.
* Cấu trúc dự án được sử dụng theo kiểu Angular.
* Sử dụng bộ định tuyến Angular để tổ chức các phần khác nhau của ứng dụng khách.

🌺🌺🌺[**Trước khi sử dụng Angular để thay đổi giao diện dự án**](https://angular.io/)🌺🌺🌺



🌺🌺🌺[**Sau khi sử dụng Angular để thay đổi giao diện dự án**](https://angular.io/)🌺🌺🌺

